WHAT IS CLAIMED IS:

1. A method for providing a virtual device container to virtually extend the functionality of a network device on a network for supporting a plurality of functional application modules residing in a server on the network, said method comprising the steps of:

receiving a function request sent from one of the functional application modules, the function request corresponding to the network device;

selecting one of a plurality of functional component modules in response to the function request, each of the functional component modules corresponding to a respective one of the functional application modules, the selected functional component module corresponding to the functional application module which sent the function request; and

executing the selected functional component module according to the function request,

wherein each functional component module communicates with the corresponding functional application module through a first interface and communicates with the network device through a second interface.

2. A method according to Claim 1, wherein each functional application module implements a different network-wide application.

3. A method according to Claim 2, wherein each functional component module provides functional support on behalf of the network device for each corresponding functional application module, respectively.

4. A method according to Claim 1, wherein one of the separate functional application modules implements an e-mail printing application.

5. A method according to Claim 1, wherein one of the separate functional application modules implements a network management application.

6. A method according to Claim 1, wherein one of the separate functional application modules implements a network security application.

7. A method according to Claim 1, wherein one of the separate functional application modules implements a resource management application.

8. A method according to Claim 1, wherein one of the separate functional application modules implements a print job accounting application.

9. A method according to Claim 1, further including the step of providing a description of each of the plurality of functional component modules for access by each of the plurality of functional application modules.

10. A method according to Claim 1, wherein the function request contains a reference to one of the plurality of functional component modules.

11. A method according to Claim 1, wherein the function request is a function call which is supported by one of the plurality of functional component modules.

12.   A method according to Claim 1, wherein the function request is supported by an operating system component interoperability standard.

5   13.   A method according to Claim 1, wherein an operating system registry contains a registry entry corresponding to each of the plurality of functional component modules.

10   14.   A method according to Claim 13, further including the step of loading, by a functional component keeper module, the functional component module corresponding to each registry entry for the network device in the operating system 15   registry.

15.   A method according to Claim 14, wherein the loading step is performed in response to an initialization command.

20

16.   A method according to Claim 1, wherein the function request is a request for information from the network device.

25   17.   A method according to Claim 1, wherein the function request is a request for the network device to receive information.

18.   A method according to Claim 1, wherein 30   the function request is a request for the network device to perform a function.

19.   A method according to Claim 1, wherein the first interface is a dedicated software bus.

35

20. A method according to Claim 19, wherein the dedicated software bus is managed by a software bus control module.

5

21. A method according to Claim 20, wherein each functional component module and each functional application module contain a software bus connector module which supports communication over the dedicated software bus.

10

22. A method according to Claim 21, wherein communication over the dedicated software bus between the software bus connector modules is implemented by using a plurality of different

15 software bus messages.

23. A method according to Claim 22, wherein one of the software bus messages is an information request from one of the functional

20 application modules for identification information corresponding to one of the functional component modules.

24. A method according to Claim 23,

25 wherein, in response to the information request, identification information corresponding to the requested functional component module is sent to the requesting functional application module.

30 25. A method according to Claim 24, wherein the requesting functional application module establishes a direct communication connection with the requested functional component module by using the identification information.

35

26. A method according to Claim 25, wherein the direct communication connection is over a data channel of the dedicated software bus.

5 27. A method according to Claim 25, wherein the direct communication connection is implemented in one of a plurality of different communication protocols.

10 28. A method according to Claim 22, wherein one of the software bus messages is an information request for identification information corresponding to all functional component modules and functional application modules that have
15 software bus connector modules.

29. A method according to Claim 22, wherein one of the software bus messages is an event message for notifying all functional component
20 modules and functional application modules of a software bus event.

30. A method according to Claim 29, wherein the software bus event is a reset event.

25

31. A method according to Claim 29, wherein the software bus event is a startup event.

32. A method according to Claim 29,
30 wherein the software bus event is a shutdown event.

33. A method according to Claim 29, wherein the software bus event is a pause event.

35 34. A method according to Claim 22, wherein one of the software bus messages is a subscription message from one of the software bus

connector modules, wherein, in response to the subscription message, the dedicated software bus subsequently passes all software bus messages of a specified type to the requesting software bus connector module.

35. A method according to Claim 22, wherein the software bus messages are supported by a plurality of different communication protocols.

36. A method according to Claim 1, wherein the second interface is the network.

37. A method according to Claim 1, wherein the second interface is a serial bus.

38. A method according to Claim 1, wherein each functional component module reads data from a memory of the network device via the second interface.

39. A method according to Claim 1, wherein each functional component module writes data to a memory of the network device via the second interface.

40. A method according to Claim 1, wherein one of the functional application modules is a proxy application which provides a data interface over the network between the plurality of functional component modules and a third-party application.

41. A method according to Claim 1, wherein the function request is a generic request which is supported in the selected functional component module by a plurality of specific protocol requests, and wherein one of the plurality of specific

protocol requests is sent from the selected
functional component module to the network device
based on a desired protocol for communication with
the network device.

5

     42.   A method according to Claim 1, wherein
a second plurality of functional component modules
are used to support a second network device, and
wherein each functional application module is

10   supported by the corresponding functional component
module for each network device.

     43.   A method according to Claim 1, wherein
the virtual device container is a DCOM server.

15

     44.   A method according to Claim 43,
wherein the function request is addressed to the
virtual device container.

20   45.   A method according to Claim 19,
wherein the virtual device container registers an
entry with the dedicated software bus, the entry
containing a node address corresponding to the
virtual device container.

25

     46.   A method according to Claim 45,
wherein the virtual device container obtains a
globally unique identifier through an operating
system function call.

30

     47.   A method according to Claim 46,
wherein each of the functional application modules
has access to the globally unique identifier from
the virtual device container by using the node

35   address, whereupon a direct software connection is
available to the virtual device container by using
the globally unique identifier.

48. A method according to Claim 45,
wherein at least one of the functional application
modules establishes a direct software connection to
the virtual device container by using the node
5   address of the virtual device container.

49. A method according to Claim 48,
wherein the direct software connection is supported
by COM function calls.

10

50. A method according to Claim 48,
wherein the direct software connection is supported
by JAVA function calls.

15   51. A method according to Claim 48,
wherein the direct software connection is supported
by CORBA function calls.

52. A method according to Claim 46,
20   wherein the globally unique identifier is obtained
from an operating system function call.

53. A method according to Claim 46,
wherein the function request is received over a
25   direct software connection between the corresponding
functional application module and the virtual device
container, the direct software connection being
established based on the globally unique identifier.

30   54. A method for providing a virtual
device container to virtually extend the
functionality of a network device on a network for
supporting a plurality of functional application
modules residing in a server on the network, said
35   method comprising the steps of:
loading, by a functional component keeper
module in the virtual device container, a plurality

of functional component modules corresponding to a plurality of registry entries in an operating system registry, each of the functional component modules corresponding to a respective one of the functional application modules;

establishing a direct connection between a requesting one of the functional application modules and the virtual device container over a dedicated software bus by using a globally unique identifier which corresponds to the virtual device container and which is obtained from the virtual device container via the dedicated software bus;

receiving, over the direct connection, a function request sent from the requesting functional application module, the function request corresponding to the network device and containing a function call;

selecting one of a plurality of functional component modules for supporting the function call, the selected functional component module corresponding to the requesting functional application module; and

executing the selected functional component module according to the function call,

wherein each functional component module communicates with the network device through the network.

55. A network computing device for providing a virtual device container to virtually extend the functionality of a network device on a network for supporting a plurality of functional application modules residing in a server on the network, comprising:

a program memory for storing process steps executable to perform a method according to any of Claims 1 to 54; and

a processor for executing the process steps stored in said program memory.

56. Computer-executable process steps stored on a computer readable medium, said computer-executable process steps for providing a virtual device container to virtually extend the functionality of a network device on a network for supporting a plurality of functional application modules residing in a server on the network, said computer-executable process steps comprising process steps executable to perform a method according to any of Claims 1 to 54.

57. A computer-readable medium which stores computer-executable process steps, the computer-executable process steps to provide a virtual device container to virtually extend the functionality of a network device on a network for supporting a plurality of functional application modules residing in a server on the network , said computer-executable process steps comprising process steps executable to perform a method according to any of Claims 1 to 54.